coursera

ⓘ This item has a late penalty. A penalty of 10% will be applied to your score per late day after the deadline. Learn More

Programming Assignment: Mining Contiguous Sequential Patterns in Text

✔ Passed · 100/100 points

Deadline   The assignment was due on February 24, 09:59 PM PST
You can still pass this assignment before the course ends.

Instructions   My submission   Discussions

Description

In this programming assignment, you are required to implement a contiguous sequential pattern mining algorithm and apply it on text data to mine potential phrase candidates.

Input

The provided input file ("reviews_sample.txt") consists of 10,000 online reviews from Yelp users. The reviews have been stemmed (to remove the postfix of each word so words with similar semantics can have the same form), and most of the punctuation has been removed. Therefore, each line is basically a list of strings separated by spaces.

An example line is provided below:

```
1   cold cheap beer good bar food good service looking great pittsburgh style
        fish sandwich place breading light fish plentiful good side home cut
        fry good grilled chicken salad steak soup day homemade lot special
        great place lunch bar snack beer
```

reviews_sample.txt

Output

You need to implement an algorithm to mine *contiguous* sequential patterns that are frequent in the input data. A contiguous sequential pattern is a sequence of items that frequently appears as a consecutive subsequence in a database of many sequences. For example, if the database is

```
1   A,B,A,C
2   A,C,A,B,A,B
3   B,A,A,C,D
4   
```

and the minimum support is 2, then patterns like "A,B,A" or "A,C" are both frequent contiguous sequential patterns, while the pattern "A,A" is not a frequent *contiguous* sequential pattern because in the first two sequences the two A's are not consecutive to each other. Notice that it is still a frequent sequential pattern though.

Also, notice that multiple appearances of a subsequence in a single sequence record only counts once. For example, the pattern "A,B" appears 1 time in the first sequence and 2 times in the second, but its support should be calculated as 2, as there are only 2 records containing subsequence "A,B".

When implementing the algorithm, you could use any programming language you like. We only need your resulting pattern file, not your source code file.

Please set the relative minimum support to 0.01 and run it on the given text file. In other words, you need to extract all the frequent contiguous sequential patterns that have an absolute support no smaller than 100.

Please write all the frequent contiguous sequential patterns along with their absolute supports into a text file named "patterns.txt". Every line corresponds to exactly one pattern you found and should be in the following format:

support:item_1;item_2;item_3

For example, suppose the phrase "parking lot" has an absolute support 133, then

the line corresponding to this frequent contiguous sequential pattern in "patterns.txt" should be:

133:parking;lot

Notice that the order does matter in sequential pattern mining. That is to say,

133:lot;parking

may be graded as incorrect.

Important Tips

Make sure that you format each line correctly in the output file. For instance, use a semicolon instead of other characters to separate different items in the sequence.

Notice that the order does matter in sequential pattern mining. That is to say,

133:lot;parking

may be graded as incorrect

even if

133:parking;lot

is a frequent contiguous sequential pattern.

Make sure you also include all the length-1 patterns.

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.